http://mvpdagen.no

#MVPdagen

Introduksjon til Infrastructure as Code i Azure

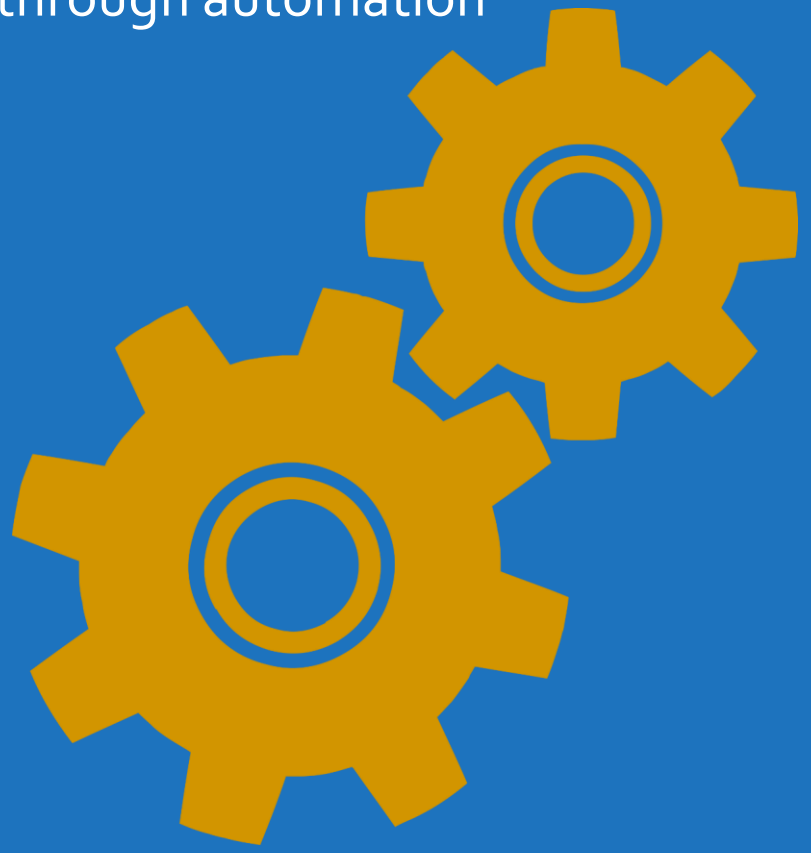
Jan Egil Ring
Lead Architect, Crayon
Azure MVP
@JanEgilRing

AGENDA

- Infrastructure as Code
  - Azure Resource Manager
  - Terraform
- Demo
  - Azure Resource Manager templates
  - Terraform fra kommandolinje
  - Terraform fra Azure DevOps
- Oppsummering

WHAT IS INFRASTRUCTURE AS CODE (IAC)

- Build the infrastructure for an application all at once through automation
- Not just for Cloud, Software Defined Data Center
- Embedded Documentation
- Source Control
- Flexible Build Process



THE JOURNEY OF AN AZURE USER

Crawl

Manually create and manage resources in **Azure Portal** or **QuickStart**

Walk

Automate deployment of Azure resources using **Infrastructure as Code**.

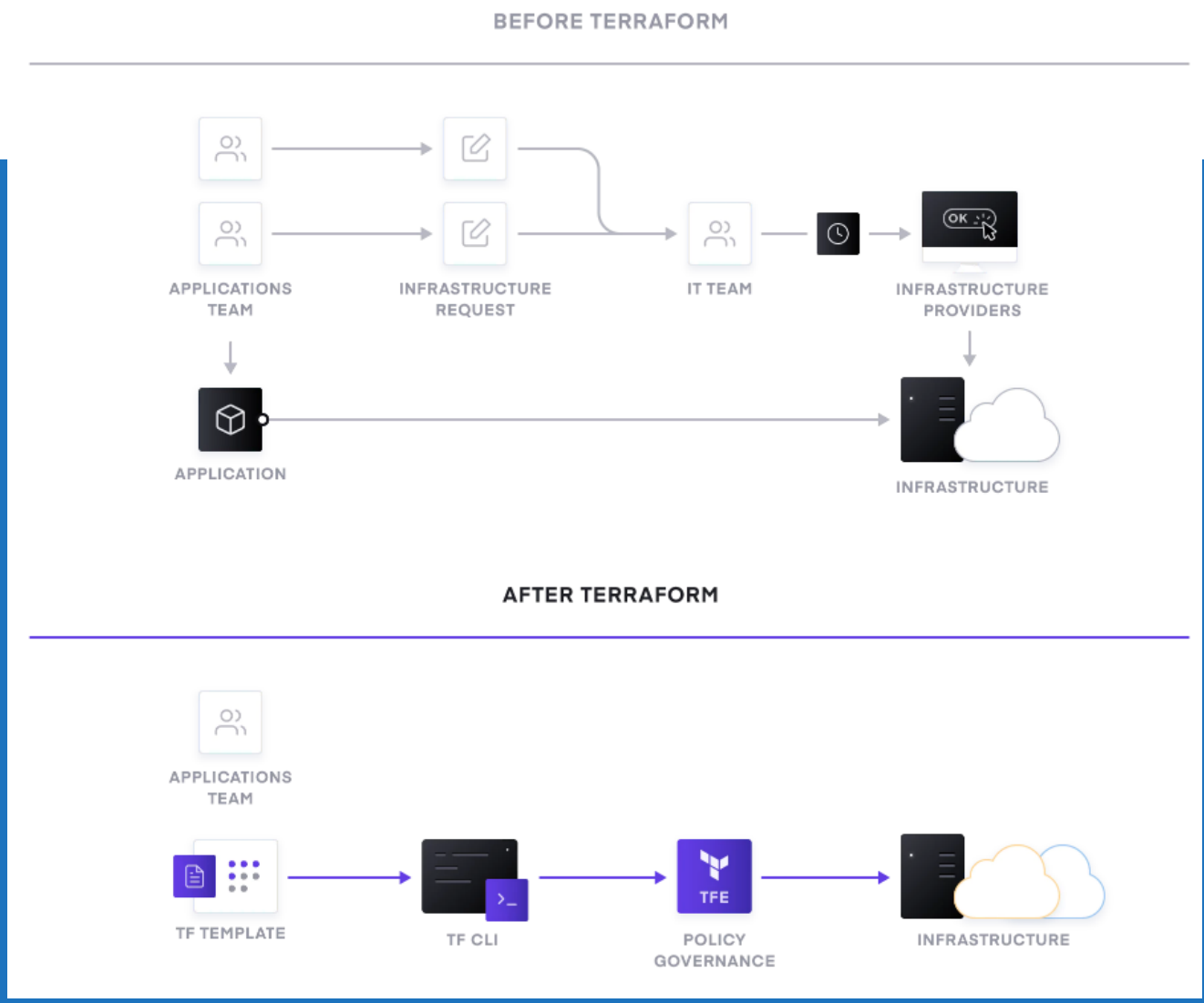
E.g. **ARM templates**, PowerShell or Terraform.

Run

Orchestrate deployment of Azure resources using CI/CD tools.

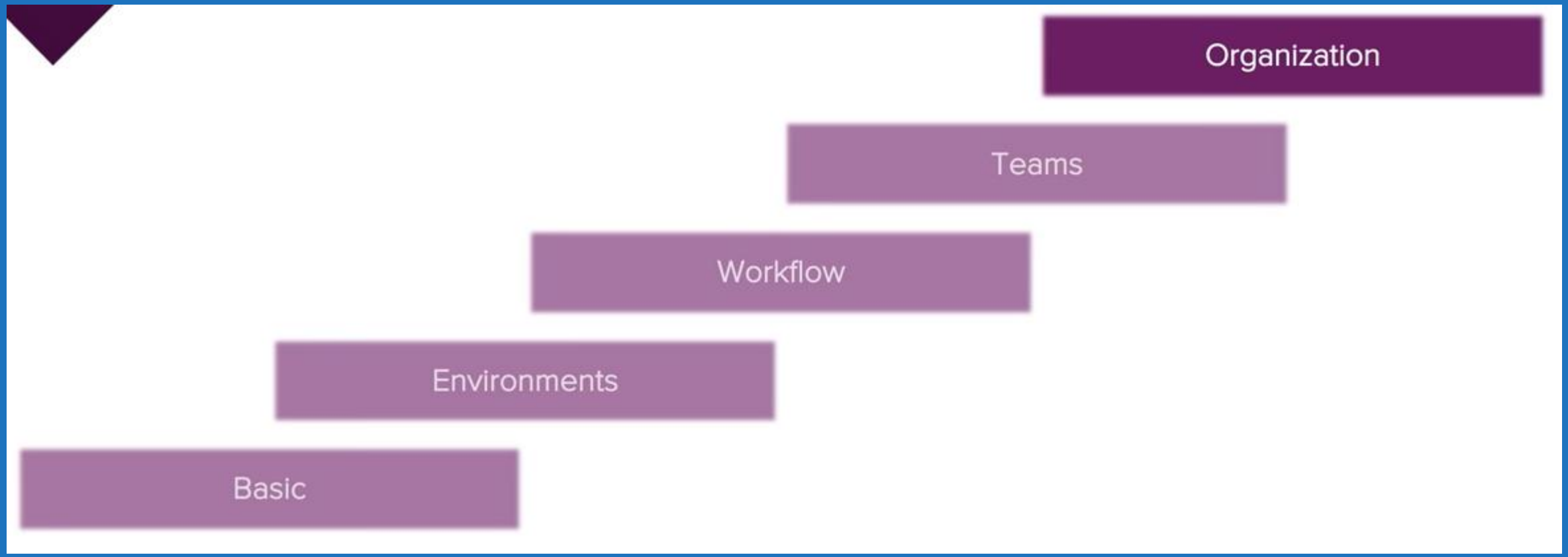
E.g. **Azure DevOps** or Jenkins.

CLOUD OPERATING MODEL



https://www.hashicorp.com/cloud-operating-model

ADOPSJON AV INFRASTRUCTURE AS CODE



https://www.youtube.com/watch?v=G06j6HLWyYo

WHY DO IAC

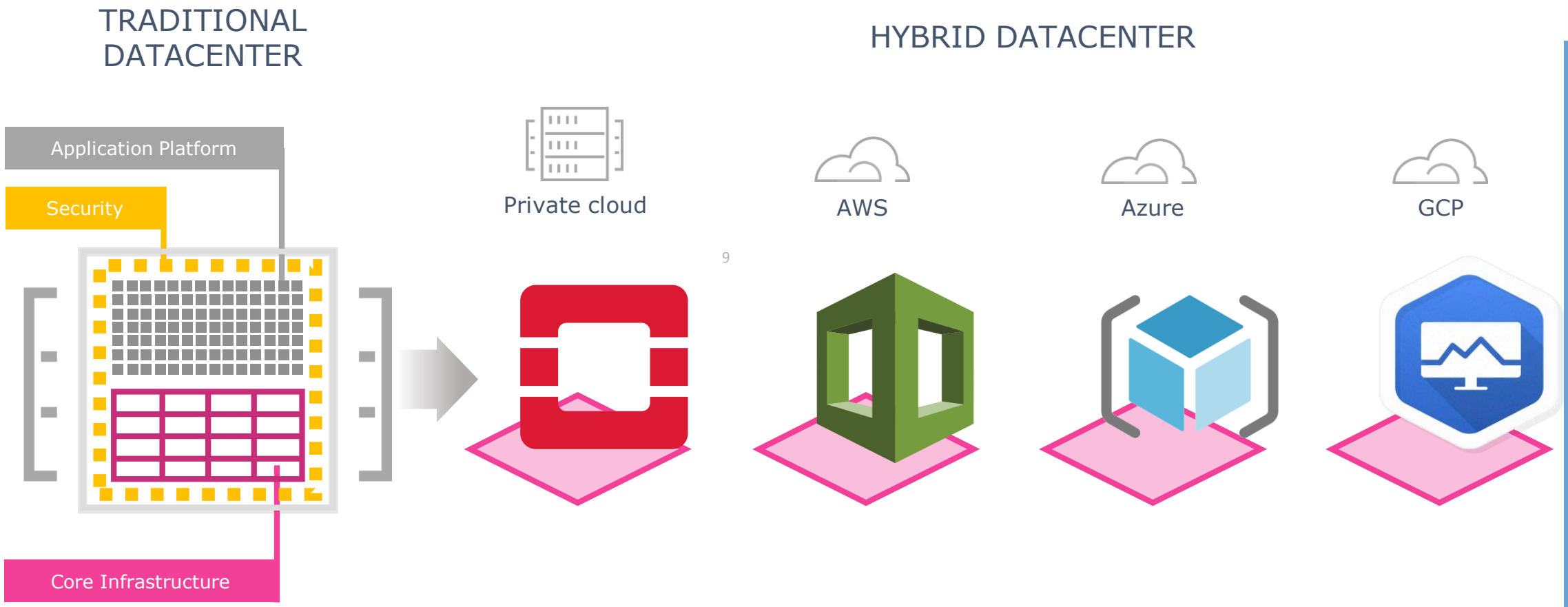
- Faster to deliver
- Flexibility
- Code is documentation

INFRASTRUCTURE AS CODE

- ☑ Reproducible Environments
- ☑ Automation – CI/ CD
- ☑ Trackable – source control
- ☑ Workflow
- ☑ Providers

☒ Apply same config across clouds

MULTI-CLOUD INFRASTRUCTURE TRANSITION



HOW TO GET STARTED

People

- Have a Vision
- This is a big change
- Iac/DevOps Movement
- A way of life
- Required to be successful

Process

- Simplicity
- Modular
- Flexible
- Versioning

Products

- PowerShell/Bash
- Azure Quickstart Templates
- VS Code
- GitHub
- Azure Automation, Ansible, Azure DevOps, Terraform

AZURE RESOURCE MANAGER TEMPLATES

- Written in JSON
- Tooling for Visual Studio and Visual Studio Code
- Native Azure portal integration
- Generated directly from REST / Swagger

AZURE RESOURCE MANAGER TEMPLATE

```
{
    "$schema": "https://schema.management.azure.com/..json#",
    "contentVersion": "1.0.0.0",
    "parameters": {},
    "variables": {},
    "resources": [{
            "type": "Microsoft.Resources/resourceGroups",
            "apiVersion": "2018-05-01",
            "location": "eastus",
            "name": "demo-storage",
            "properties": {}
        },
        {
            "type": "Microsoft.Storage/storageAccounts",
            "name": "demo-storage",
            "apiVersion": "2018-02-01",
            "location": "eastus",
            "sku": {
                "name": "Standard_LRS"
            },
            "kind": "Storage",
            "properties": {}
        }
    ]
}
```

Resource Group

Storage Account

@nepeters

DEMO

Azure Resource Manager templates

TERRAFORM

- Open source project
- Cross computing environment templating language
- Provision, Update, and Delete resources
- Authored in HashiCorp Configuration Language (HCL) or JSON

TERRAFORM EXAMPLE

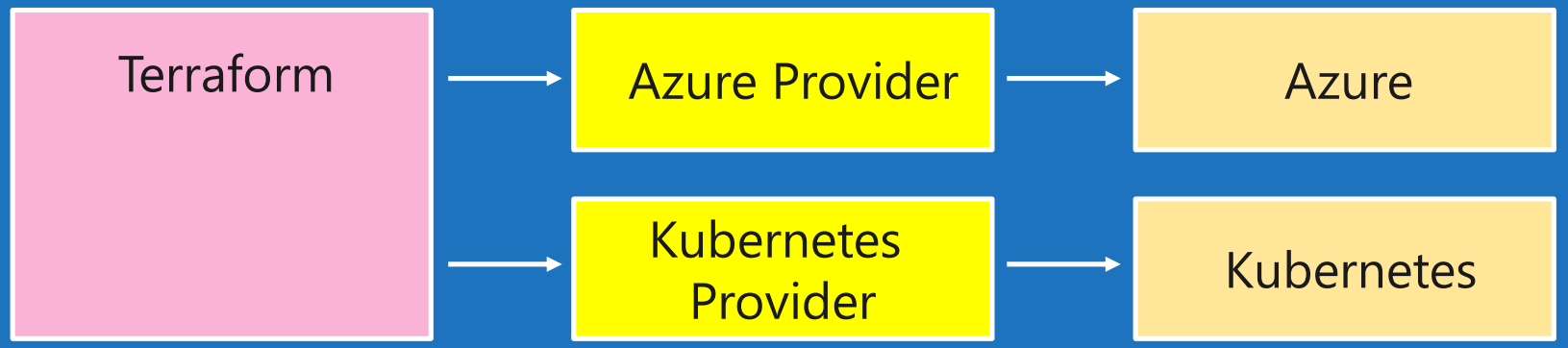
```
resource "azurerm_resource_group" "testrg" {
    name = "resourceGroupName"
    location = "westus"
}

resource "azurerm_storage_account" "testsa" {
    name = "storageaccountname"
    resource_group_name = "testrg"
    location = "westus"
    account_tier = "Standard"
    account_replication_type = "GRS"
}
```

Resource Group

Storage Account

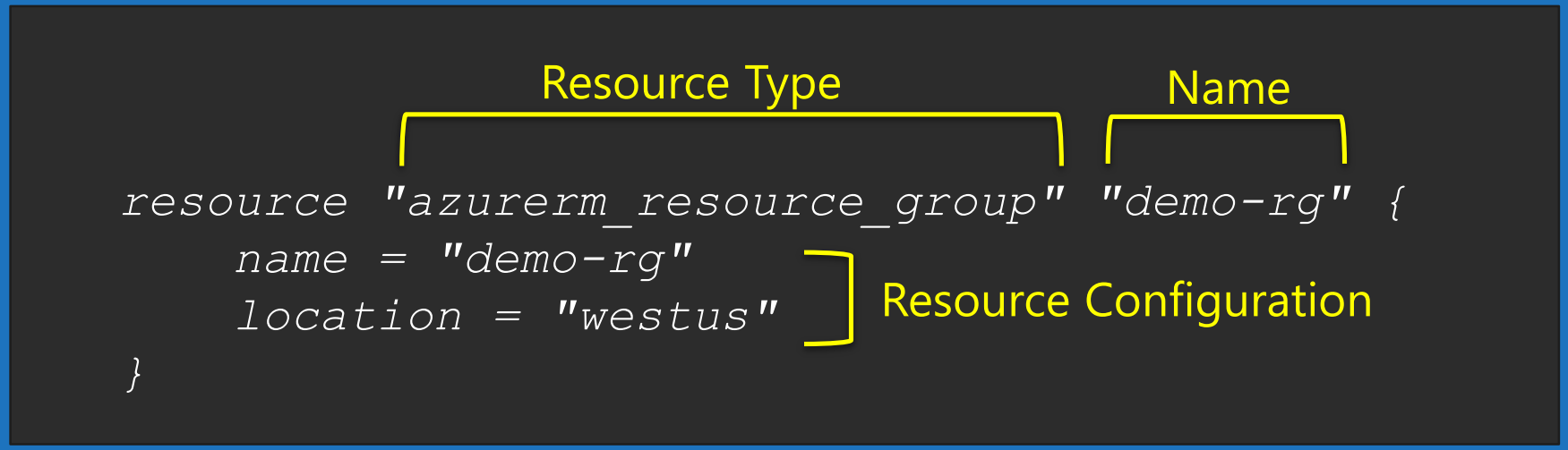
PROVIDERS

- Terraform 'extensions' for deploying resources
- Manages cloud / endpoint specific API interactions
- Available for major clouds and other platforms
- Hand authored (azurerm)



BASIC RESOURCE CREATION

- Resource Type: required provider
- Name: internal name
- Configuration: deployment details

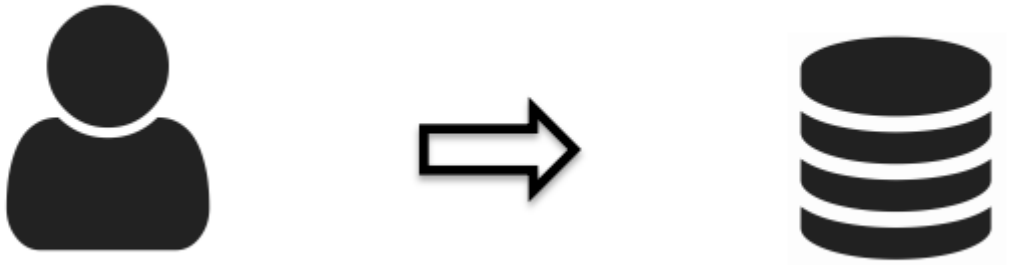


BASIC TERRAFORM COMMANDS

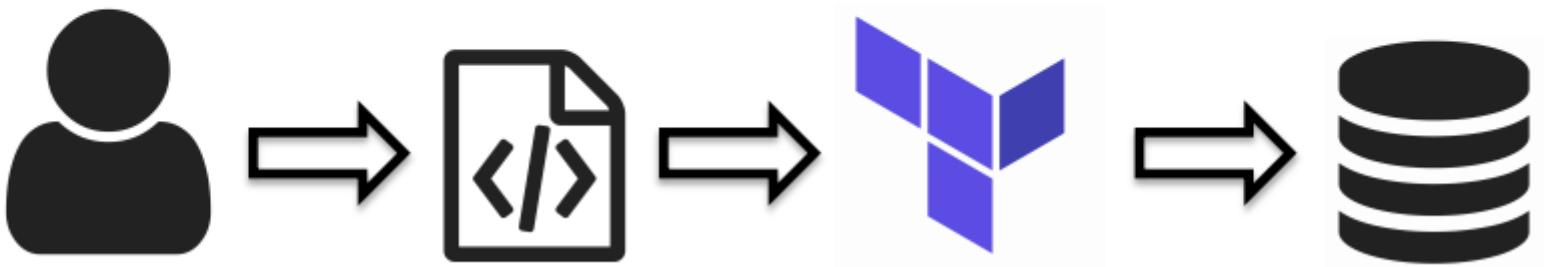
- Terraform init – initializes working directory
- Terraform plan – pre-flight validation
- Terraform apply – deploys and updates resources
- Terraform destroy – removes all resources defined in a configuration

DEMO

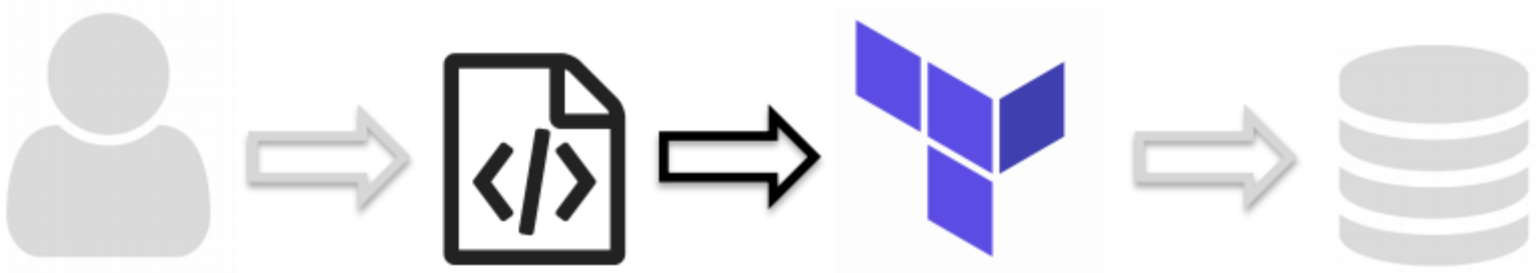
- Terraform fra kommandolinje
- Terraform i Azure DevOps



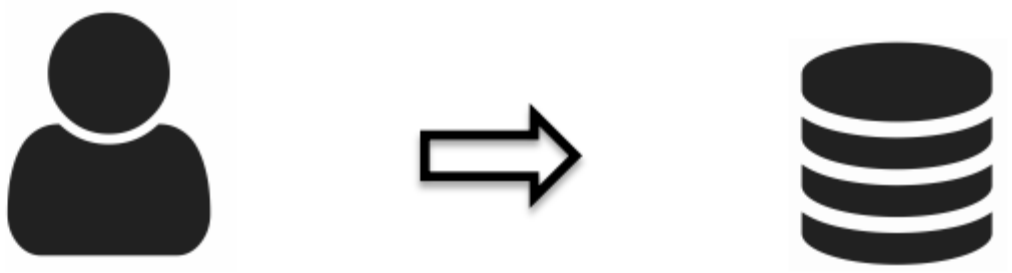
Old way: make changes directly and manually



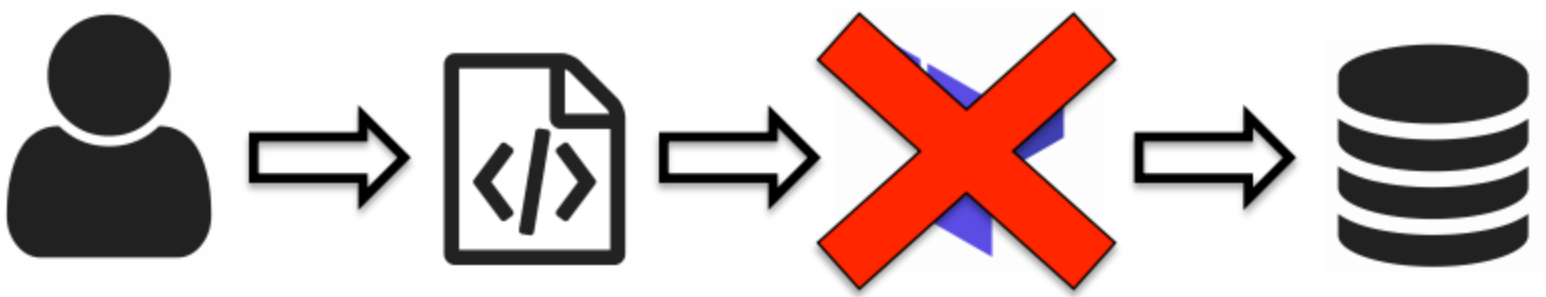
New way: make changes indirectly and automatically



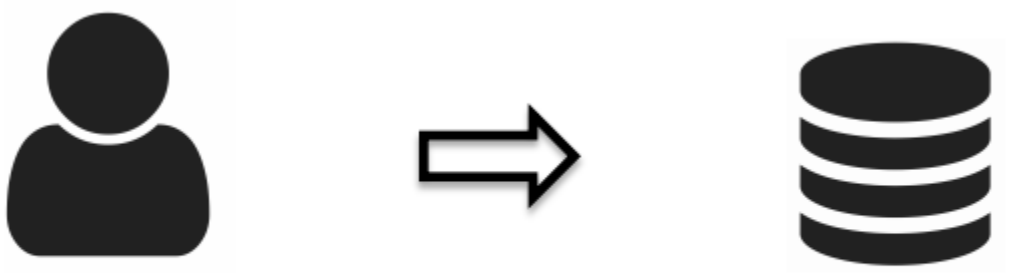
Learning these takes time



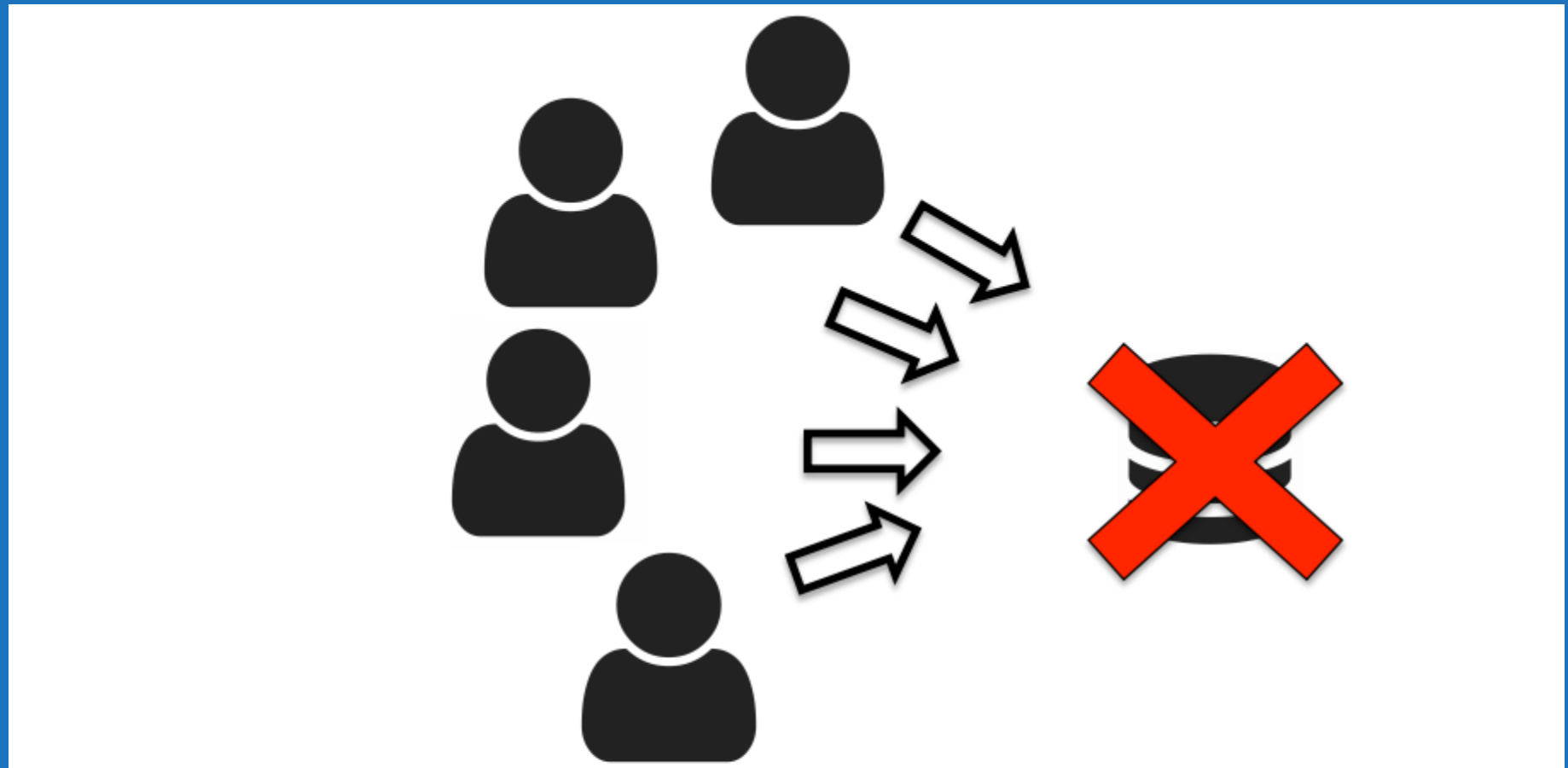
More time than making a change directly...



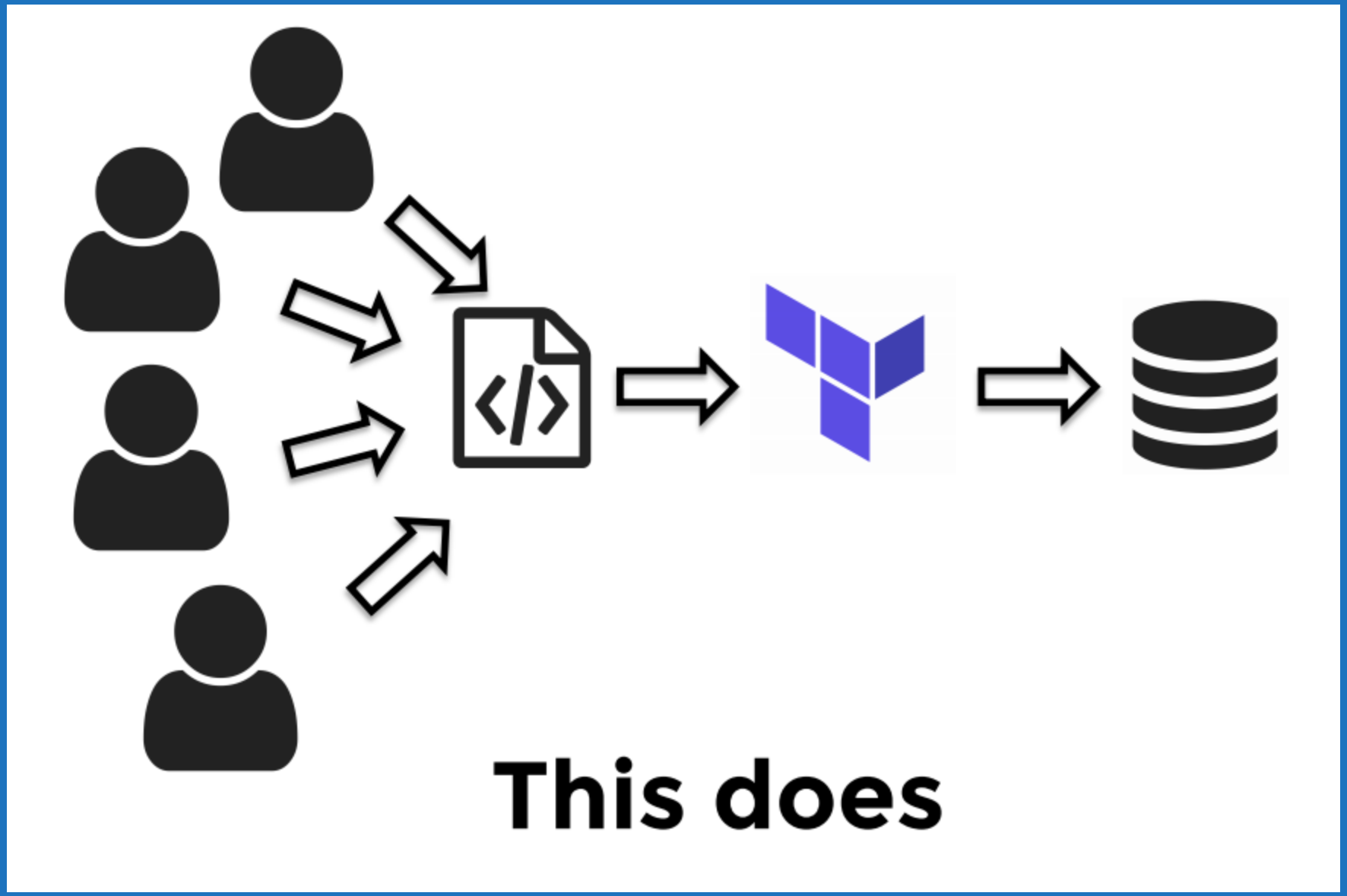
And the next person to try to use it will get errors



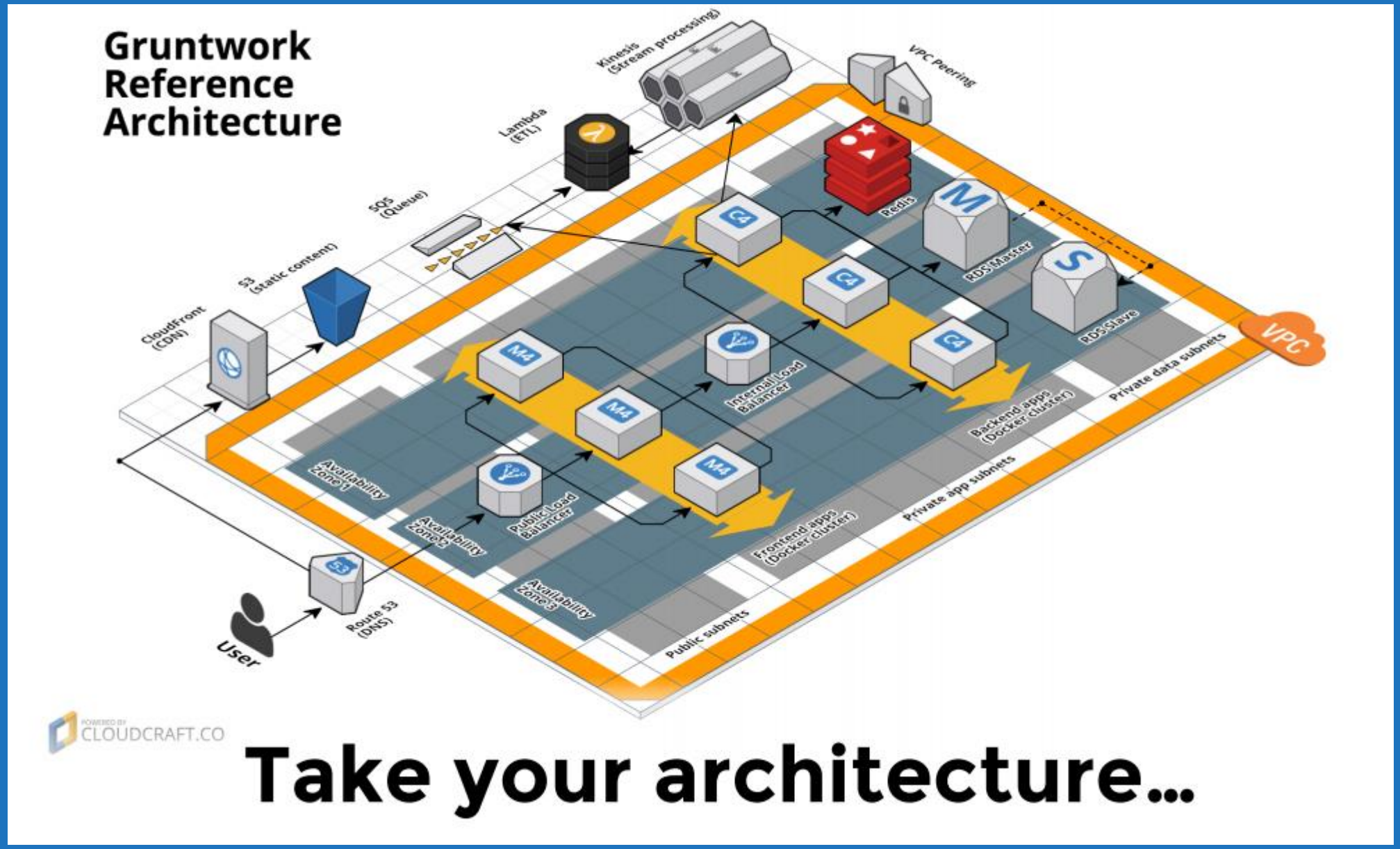
So then they'll fall back and make manual changes



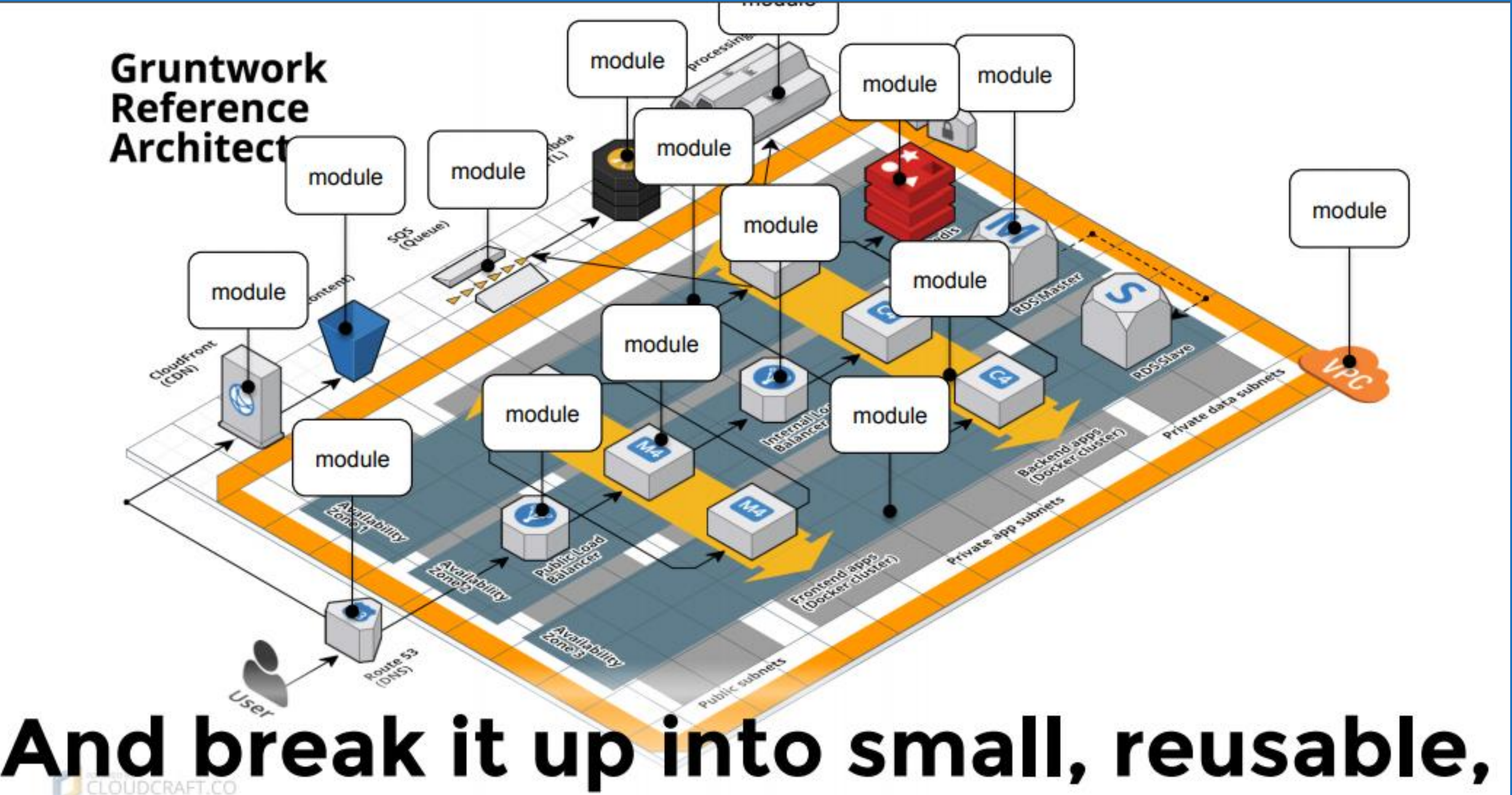
But making manual changes does not scale


This does

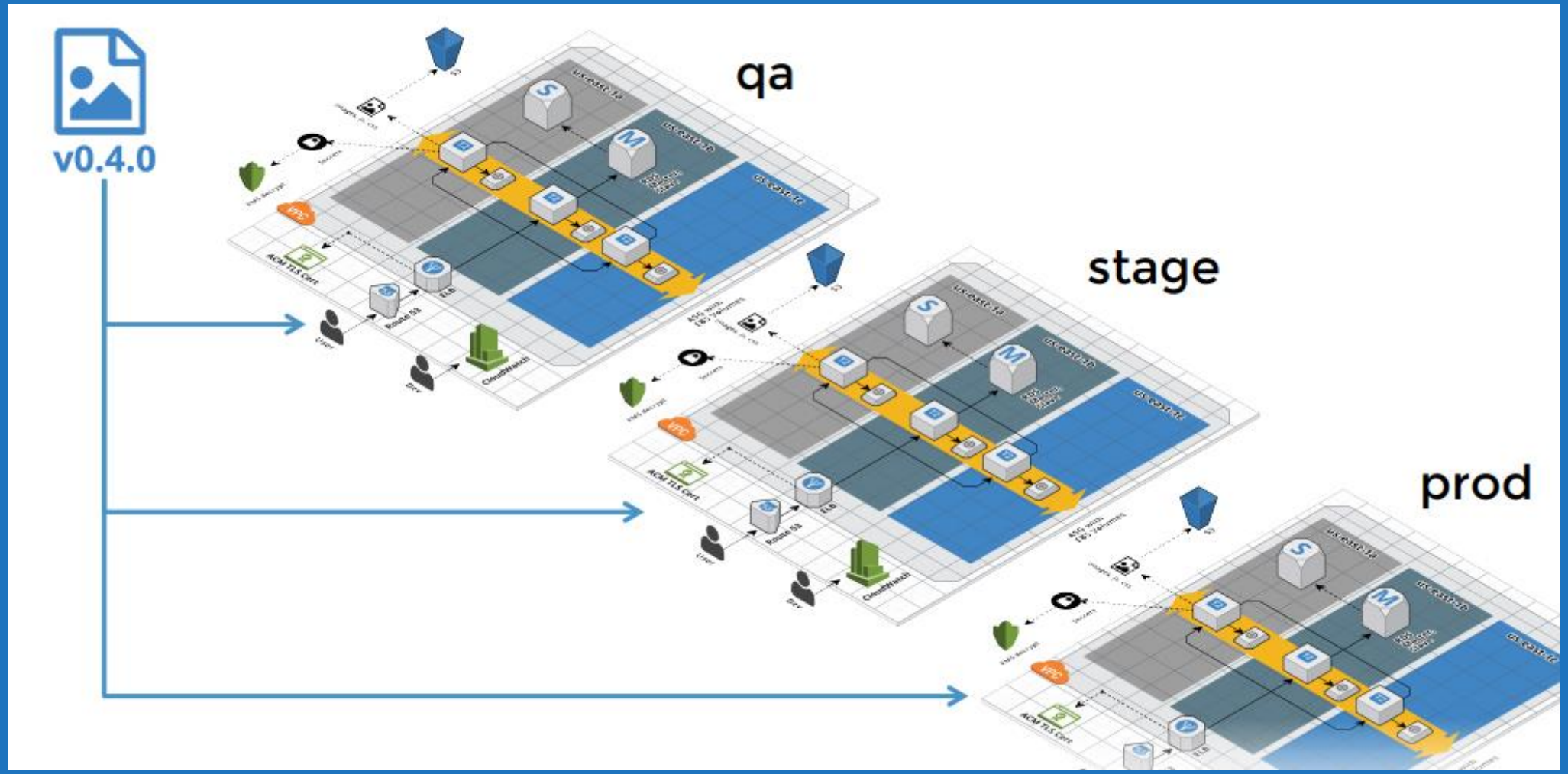
Key takeaway: tools are not enough. You also need to change behavior.



Take your architecture...



And break it up into small, reusable, standalone, tested modules


v0.4.0
qa
stage
prod

Tusen takk for meg!

#MVPdagen

http://mvpdagen.no

RESSURSER

Terraform – nedlasting og dokumentasjon
www.terraform.io